Proceedings of the 38th
Conference on Decision & Control
Phoenix, Arizona USA • December 1999

Demand-Driven, Iterative Capacity Allocation and Cycle Time Estimation for Re-entrant Lines*

Shi-Chung Chang
Control and Decision Laboratory
Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.
Email: scchang@cc.ee.ntu.edu.tw

Abstract *– Daily production target setting for each production stage of a semiconductor wafer fabrication factory (fab) is a challenging machine capacity allocation problem due to the complex and re-entrant process flows. This paper summarizes a methodology developed by the author and his colleagues for the design of a daily target setting system (TSS) over the past few years. The methodology realizes PULL-then-PUSH and proportional capacity allocation principles to meet production demands in a smooth way while maximizing machine utilization. As machine capacity allocation and available wafer flows are intertwined, the target setting problem can be viewed as a fixed-point iteration problem. A deterministic queueing analysis-based algorithm is designed to estimate cycle times and hence wafer flows. The methodology iterates between capacity allocation and cycle time estimation until a fixed-point capacity allocation is achieved. Recorded field implementation results of various versions of TSS based on the methodology include a decline of average total WIP by 8%, an increase of daily total wafer flows by nearly 20%, an increase in target achieving rate at the shop floor by more than 10%, the reduction of average cycle time per layer from 3.25 to 2.96 days , and the reduction of average +2 sigma of per-layer cycle time from 4.63 to 3.68.*

I.Introduction

Production scheduling of semiconductor wafer fabrication plants (fabs) has drawn much research attention due to its technical challenges as well as significance. The fabrication process of each type of wafers may require more than three hundred fabrication steps; the whole process involves tens of delicate and expensive equipment groups. The process flow of a type of wafers through the fab is highly reentrant because the wafers make multiple visits to equipment groups as successive circuit layers are added onto the wafer. This reentrant feature poses a unique challenge to production scheduling, since wafers of different part types as well as wafers of the same type but at different layers of fabrication may compete for the finite capacity of an equipment group.

Besides, there are multiple production objectives for scheduling a fab such as

- (o1) to meet production targets such as monthly total output volume and master production schedule (MPS),
- (o2) to maximize total number of moves achieved per day;
- (o3) to balance the production line,
- (o4) to reduce wafer-in-process (WIP) and cycle time in both average and variance,
- (o5) to maximize machine utilization, and
- (o6) to facilitate on-time delivery.

The emphasis among these objectives may vary over time based on market demands and internal management strategies.

Just like in other complex production systems, hierarchical scheduling is widely adopted by fabs based on the time-scale of production flow control (PFC) activities. A frequently used, three-level decision hierarchy of scheduling functions by fabs are (1) wafer release and output scheduling, (2) daily target setting, and (3) machine assignment and lot dispatching. Wafer release and output scheduling aims at controlling the WIP level and cycle time while meeting delivery requirements. It schedules the quantities of wafer release and output for a fab using a day as a time unit over two to four months. Machine assignments and lot dispatching assigns machines in each equipment group to various production steps requiring the same type of machines in a time unit of a few (3-8) hours over one day and ranks lot priorities of processing in the same step. Daily target setting combines the release/output schedule, present fab production states, and machine capacity forecast to properly set target volumes of daily production for individual part types and steps. These daily targets then serve as a guideline for machine assignment and lot dispatching. Daily target setting thus bridges between fab wide and local PFC activities. Among the three PFC functions just described, this paper will focus on daily target setting.

In the literature, there has been active research on scheduling daily operations of fabs. Wein [Wei88] conducted simulation study of PFC and reached at a conclusion that wafer start policy impacts on fab performance more than dispatching rules. Lu et al [LRK94] rebutted such a view point and identified two best dispatching policies which can significantly reduce both the mean and the variance of cycle times. Glassey et al. [GSS96] developed a hybrid simulation-optimization procedure to delineate linear control rules for fabs. Conors et. al. [CFY94] adopted fluid network models for scheduling operations of high volume fabs. The minimum inventory variability scheduling (MIVS) scheme proposed by Li et al. [LTC96] calculates a reference WIP level for each step based on output demands and determines the priority among steps by comparing the actual and the reference WIP distributions. Liao et al. [LCP96] adopted an integer programming problem formulation and designed for a pilot line an Lagrange relaxation and network flow-based daily scheduling tool, ERSOFS. There have also reports on scheduling practices across various semiconductor fabrication companies [Lea94]. They are mostly customized designs, involving some commercially available modules, Kanban Logic, cycle time tracking, rule-based system, deterministic simulation, etc.

This paper summarizes the methodology developed by Chang et. al. for the design of a daily target setting system (TSS) over the past few years [CLP95, CCW95, WWT98]. The remainder of the paper is organized as follows. Section 2 first gives an overview of TSS. Section 3 then presents the main body of the iterative, proportional resource allocation approach. A cycle time estimation algorithm is described in Section 4. Section 5 presents the three-phase field implementation results. Finally, Section 6 concludes the paper.

*This work was supported in part by the National Science Council of the Republic of China and Taiwan Semiconductor Manufacturing Co. under Grants NSC 85-2622-E-002-018R and NSC 86-2622-E-002-025R.

II. Problem Description and System Overview

Consider a fab that fabricates various types of circuitry on silicon wafers. Every product has its own process flow, i.e., a sequence of fabrication steps. Processing machines in the fab are of diversified characteristics and failure prune and are divided into groups according to their functionality. Each fabrication step requires a machine from a specific group for processing. Due to the layered nature of fabricating circuitry on a wafer, a few fabrication steps of one process flow may use machines from one same group, i.e., a wafer may make multiple visits to a machine group in its fabrication sequence. There exist many process and operation constraints.

To focus on key issues of daily target setting and to reduce problem complexity to a comprehensible level, we make a few assumptions and simplifications as follows.

A1. A schedule of wafer release and output over the coming three months is available from MPS.

A2. Detailed characteristics and constraints of machine operations are neglected; expected machine capacity in terms of number of wafers per day are given and used as an aggregate characterization.

A3. There exists no priority difference among product types.

A4. The unit of fabrication at each machine is a wafer and the processing time per wafer at a batch processing machine is approximated as the batch processing time divided by the average number of wafers in a batch.

A5. The fabrication process of each product type is fixed.

A6. Stage is adopted as the basic management unit for describing process flows, where a stage of a product type is obtained by aggregating a few consecutive fabrication steps of the product type. A completion of the fabrication of a stage for a wafer is called a *move*.

A7. Each stage has a corresponding *key machine group,* which is the machine group for processing the bottleneck step of the stage. The bottleneck step of a stage can be defined either empirically or as a step that has the highest ratio of processing time per wafer over number of machines in the machine group required by the step.

A8. All the stages of various fabrication processes are arranged into a global sequence of totally J stages in a way that if stage k precedes a stage k' in one process flow, then stage k also proceeds stage k' in the global stage sequence.

A9. Buffer space is large and can be considered as infinite.

A10. Machine service discipline at each stage is first-come-first-serve (FCFS).

A11. Wafer transfer times among machines are negligible.

Inputs of the target setting problem include

- process flows of individual product types in terms of stages,
- key machine group to each stage,
- wafer-in-process (WIP) levels by stage by product type,
- wafer output quantity to date by product type,
- stage processing or cycle (waiting plus processing) times by product type, and
- MPS (wafer start and output) for the coming three months.

In the practice of fab operations, daily targets are set under a mixed consideration of the objectives (o1) - (o6) as described in the Introduction with the specific emphasis dynamically changing according to market demands and business strategies of the fab. The daily target setting problem is then to determine for each product type the target number of moves to be achieved at each stage during one day based on the aforementioned inputs and a mixed consideration of opreation objectives. It is essentially equivalent to a problem of how to allocate machine capacities of one day to the processing of individual product types at various stages. Complexity of the problem is basically due to the large numbers of product types, fabrication stages, machines and operation constraints. Further challenges of the target setting problem are rooted in machine and production uncertainties, the re-entrant nature of process flow, meeting the MPS and the need for adapting to changes in operation objectives in a fab.

A Target Setting System (TSS) [WWT98] is designed as a computer aided decision support for daily target setting. It includes three modules: (1) Master Production Schedule Conversion (MPSC), (2) Capacity Estimation(CE), and (3) Target Setting Main Function (TSMF). The MPSC module is designed to be run daily over a time horizon of three months. It exploits the output schedule of MPS, current WIP distribution and actual output to-date to determine daily demanded number of moves for each type of products at all its production stage over the next 90 days. The term "number of moves" sometimes will be referred to as "moves" in short thereafter. The daily demanded moves for a stage can be interpreted as daily production orders to the stage. The MPSC module therefor serves as a hook for integration between mid-term target setting (MPS) and short-term (daily) target setting. Ideally, if all demanded moves from MPSC are achieved daily, on-time delivery of MPS demands can be accomplished and cycle time variation controlled. The CE module estimates the available capacity in the unit of moves per day for each machine group. It is obvious that results of target setting are sensitive to capacity estimates. However, the number of moves that can be accomplished by one machine group in one day is a function of product mix, machine availability, WIP distribution and production scheduling/ dispatching policy and varies from day to day. And finally, the target setting main function (TSMF) takes daily stage demands from MPSC, capacity estimates for individual machine groups from CE, and WIP distribution from the manufacturing execution system (MES) to calculate the daily target moves by product type and by stage. Note that capacity allocation (target settings) to individual stages affect wafer flows and WIP distribution, and the vise versa. The remainder of this paper will focus on describing the methodology for realizing TSMF.

III. Target Setting Main Function

Given the demanded moves, capacity estimates and available WIP distribution as inputs, TSMF adopts an iterative scheme for determining target moves of individual stages and the corresponding WIP distribution if the targets are actually achieved. Key ideas consist of PULL, PUSH, proportional machine capacity allocation and cycle time/wafer flow-in estimation. To describe key ideas of the TSMF algorithm, let us first define some notations.

Notations

J: total number of stages;

j: stage index, j=1,...,J;

T: time horizon;

t: time period (day) index, t=1,...,T;

d: current day;

I: total number of part types;

I: part type index, I=1,...,I.

$WIP_{ij}(t)$: type-i WIP level of stage j at the beginning of day t, j=1,...,J;

$MPS\_day\_demand_{ij}(t)$: total demanded type-i moves of day t, including the planned number of moves of day t and the delayed number of moves up to the end of day t-1.

Tom_Dem_{ij}: daily MPS demand for type i at stage j in the next day;

WIP_{ij}: the WIP level of type-i parts at stage j at the beginning of the current day;

Ref_WIP_{ij}: the desirable type-i WIP level at the beginning of next day production for stage j.

$Flow\_\ in_{ij}$: number of type-i wafers flowing to stage-j from its up-stream stages during the day;

CT_{ij}: average time that a wafer spends at stage j, starting from its arrival to its departure;

M: total number of machine groups;

m: machine group index, m=1,...,M;

$C_m(t)$: estimated capacity of machine group m in term of number of moves per day;

S_m: the set of all part type and stage pairs that require machine group m for processing;

Decision Variable

$Target_{ij}$: number of type-i wafers leaving stage j to its down-stream stages during the day.

In the practice of shop floor control, WIP levels are easily measurable indications of operational health. In addition to move targets, practitioners often desire that a reference WIP distribution be given as one of the guidelines for their daily production. In this paper, the reference WIP level of type-i wafers at a stage is defined as the average WIP level under MPS demanded type-i moves for the next day, which can be computed by Little's formula as

$$Ref\_WIP_{ij} = Tom\_Dem_{ij} * CT_{ij}. \qquad (3.1)$$

Such a definition is intended to guide production efforts of the current day to a desirable WIP distribution for next day operation.

Priority of daily production effort is first given to catching up with the MPS. Consider type-i wafers at stage j. Let stage j+1 be the down stream stage to stage j+1 in the production process of type-i wafers. The number of moves that needs to be pulled from stage j by stage j+1, or equivalently, the number of wafers that stage j should supply to stage j+1, in order to fulfill the demanded type-i moves at stage j+1 is

$$Pulld_{ij}, = Max(0,\ MPS\_day\_demand_{i(j+1)} + Ref\_WIP_{i(j+1)} - WIP_{i(j+1)}\ ), \qquad (3.2)$$

Besides its supply to the down stream stage, stage j has also its own daily demanded moves to fulfill. The maximum of the two numbers are taken to reflect an aggressive attempt in catching up with the MPS schedule and set

$$Pulls_{ij}, = Max\ (MPS\_day\_demand_{ij},\ Pulld_{ij}). \qquad (3.3)$$

What can possibly be pulled at stage j must not exceed what is available from the initial wafers and wafers that may flow into the stage during the day. So the "pull force" for type-i part at stage j is

$$Pullf_{ij}, = Min(Flow\_in_{ij} + WIP_{ij},,\ Pullf_{ij}). \qquad (3.4)$$

Machine capacities are then allocated among stages and part types according to their pull forces. When the total of pull forces that require one same machine group for processing does not exceed the capacity of the group, these pull forces are set as pull targets to the corresponding part types and stages. Otherwise, capacity of the machine group is proportionally allocated according to the pull forces. In summary, the pull target for each part type i that uses machine group m at a stage j is set as

$$Pull_{ij} = \begin{cases} Pullf_{ij}, \text{ if } \sum\limits_{i=1}^{I} \sum\limits_{\substack{j'=1 \\ (i,j') \in S_m}}^{J} Pullf_{ij'} \le C_m, \\ Pullf_{ij} \times \dfrac{C_m}{\sum\limits_{i=1}^{I} \sum\limits_{\substack{j'=1 \\ (i,j') \in S_m}}^{J} Pullf_{ij'}}, \text{ otherwise.} \end{cases} \qquad (3.5)$$

The pull force calculation and capacity allocation steps constitute a PULL procedure, which is clearly designed for on-time delivery. The proportional machine allocation is designed to balance the production in a fab[CCW95].

After the PULL procedure, to increase machine utilization and moves and to reduce cycle times, residual machine capacities are allocated to individual stages and part types according to the numbers of WIPs remained for processing. This is called a PUSH procedure. The residual type-i WIP at stage j after PULL is considered a "push force" and is calculated by

$$Pushf_{ij} = Max[0,\ Flow\_in_{ij} + WIP_{ij} - Pull_{ij}]. \qquad (3.6)$$

The residual capacity of a machine group m is

$$\bar{C}_m = C_m - \sum_{i=1}^{I} \sum_{\substack{j'=1 \\ (i,j') \in S_m}}^{J} Pull_{ij'}. \qquad (3.7)$$

The residual capacity of machine group m is allocated among stages and part types that require the group for processing according to their respective push forces the same way as that in PULL, i.e.,

$$Push_{ij} = \begin{cases} Pushf_{ij}, \text{ if } \sum\limits_{i=1}^{I} \sum\limits_{\substack{j'=1 \\ (i,j') \in S_m}}^{J} Pushf_{ij'} \le \bar{C}_m, \\ Pushf_{ij} \times \dfrac{\bar{C}_m}{\sum\limits_{i=1}^{I} \sum\limits_{\substack{j'=1 \\ (i,j') \in S_m}}^{J} Pushf_{ij'}}, \text{ otherwise.} \end{cases} \qquad (3.8)$$

The target number of type-i moves at stage j is then

$$Target_{ij} = Pull_{ij} + Push_{ij}. \qquad (3.9)$$

Note that the time delays of wafer flows have not been accounted for in our algorithmic developments so far. This issue can be easily understood by considering a fab with zero initial WIPs and a release of only one wafer at the beginning of a day. Under the aforementioned capacity allocation, targets would be set in a way that the single wafer released finishes all the fabrication stages within one day because it has the needed machine capacities at all stages. This is obviously not reasonable because the total processing time, i.e., without waiting time, of a wafer may readily exceed one day. Although time delays are clearly important to be taken into the consideration of daily target setting, how to predict their values daily is quite challenging because time delays for wafers to flow through individual stages, also called stage cycle times, depend on both daily WIP distribution and machine capacity allocation.

To incorporate time delays into daily target setting and to facilitate easy implementation, an approximate upper bound is set to limit the target moves at each stage by utilizing empirical average cycle times$\{CT_{ij}\}$and the initial WIP distribution $\{WIP_{ij}\}$. It first approximates the maximum number of type-i wafers that may possibly flow into a stage j during 24 hours by the sum of initial wafers in stages that are within one-day cycle time upstream to stage j, namely,

$$Max\_flow_{ij} = \sum_{j' \in A_j} WIP_{ij'} + \frac{(24 - \sum\limits_{j' \in A_j} CT_{ij'})}{CT_{ij*}} WIP_{ij*}, \text{ where}$$

$$A_j = \{\ j' |\ 1 \le j' < j,\ \sum_{k=j'}^{j-1} CT_{kj'} < 24\text{hrs}\} \text{ and } j* = \min A_j - 1. \qquad (3.10)$$

The target type-i moves wafers at stage j under a given set of machine capacity allocation is then further limited by Max_flow_{ij}, i.e., Eq. (3.9) is modified as

$$Target_{ij} = Min\ (Pull_{ij} + Push_{ij},\ Max\_flow_{ij}). \qquad (3.11)$$

Setting $\{Max\_flow_{ij}\}$ as upper bounds to $\{Target_{ij}\}$ thus partly captures the time delay effect of wafer flows among stages.

In both PULL and PUSH procedures, available (initial plus flow-in) WIPs of individual stages during a day affect machine capacity allocation and therefore the setting of daily targets (Eqs (3.4) and (3.6)). In turn, as the number of wafers that may flow into stage j equals to the move target of stage j-1, i.e.,

$$\mathrm{Flow\_in}_{ij} = \mathrm{Target}_{i(j-1)}, \qquad (3.12)$$

the value of $\mathrm{Flow\_in}_{ij}$ depends on machine capacity allocation of the upstream stages of stage j (Eqs. (3.5), (3.8) and (3.9)). Simply speaking, machine capacity allocation and calculation of $\{\mathrm{Flow\_in}_{ij}\}$ are just like a "chicken-and-egg" problem. Technically, the problem is a fixed-point problem, where one wants to find the values of $\{\mathrm{Target}_{ij}, \forall i, j\}$ such that if $\{\mathrm{Flow\_in}_{ij}, \forall i, j\}$ are set according to Eq. (3.10) by using these values as inputs, the sequential calculations of Eqs. (3.1) – (3.9) for all part types and stages result in output values of $\{\mathrm{Target}_{ij}, \forall i, j\}$ the same as those are input.

An Jacobi type of iterative scheme [BuF93] is designed to find the fixed-point target settings. One round of sequential application of the PULL procedure, PUSH procedure and flow-in calculation constitutes a TSMF iteration. The iterative scheme starts by setting initial values of flow-in wafers for all part types and for all stages but stage 1 to zero and setting the values for stage 1 of individual part types to their respective planned wafer release values. TSMF then calculates target settings and the corresponding flow-ins in one iteration, sets the flow-ins as inputs to start the next iteration, and repeats such iterations until the target settings calculated do not change between two successive iterations. Figure 3.1 gives a flow chart of TSMF.

IV. Stages of Penetration E+stimation Algorithm (SOPEA)

In this Section, an algorithm, SOPEA, is proposed to estimate how many stages that initial WIPs of type-i at a stage may go through in one day by utilizing processing time data, daily data of machine capacity allocations and initial WIP distribution, and deterministic queueing analysis. This algorithm in turn provides a better daily estimation of wafers flow-ins for individual stages than the empirical one in the previous Section.

Assume that $\{\mathrm{Target}_{ij}\}$ is given and that processing type-i wafers at stage j requires machines from group m. Let the number of machines allocated to processing type-i wafers at stage j be

$$n_{ij} = \mathrm{Target}_{ij}/C_m. \qquad (4.1)$$

The number n_{ij} can be given a unit of machine-day and interpreted as the machine-time allocated. So it need not be an integer. Once machine capacities are allocated to individual production flows, production flows of different part types are essentially independent of each other. The following development of a deterministic queuing analysis is therefore focused on a single type of wafer flow and the part type index i is omitted from the derivations for simplicity of presentation.

Consider the production flow between stage j to stage k, j=1,..., J-1, k=2,...,J and k>j, as shown in Figure 4.1. Denote

τ_j: processing time (in the unit of hour) at stage j

T_{jk}: the cycle time needed for the last piece of the initial WIP at stage j, WIP_j, to finish processing at stage k.

SOPEA is a recursive algorithm for computing T_{jk} by using $T_{j(k-1)}$ and $T_{(j+1)k}$. It uses the calculation of cycle time for going through two stages, i.e., k=j+1, as an algorithmic building block.

Two-Stage Cycle Time Calculation (k = j+1)

Let $\mathrm{WIP}_j\tau_j/n_j$ be the workload per machine assigned due to the initial WIP at stage j. Wafers completed at stage j go to stage j+1 and become part of the workload at stage j+1. Based on the relative workloads per machine assigned between the two stages, there are two cases for the calculation of cycle time $T_{j(j+1)}$.

Case 1: $\mathrm{WIP}_j \tau_j/n_j > [(\mathrm{WIP}_j - 1) + \mathrm{WIP}_{j+1}]\tau_{j+1}/n_{j+1}$

In this case, by the time the last piece of the WIP_j wafers is completed at stage j, all the wafers before it, i.e., $(\mathrm{WIP}_j-1) + \mathrm{WIP}_{j+1}$ pieces of wafers, finish their processing at stage j+1. In other words, this last piece of wafer does not have to wait when it arrives at stage j+1. The time that all the WIP_j pieces of wafers need to finish both stages j and j+1 is therefore

$$T_{j(j+1)} = \mathrm{WIP}_j\tau_j/n_j + \tau_{j+1}/n_{j+1}. \qquad (4.2)$$

Case 2: $\mathrm{WIP}_j \tau_j/n_j \leq [(\mathrm{WIP}_j - 1) + \mathrm{WIP}_{j+1}]\tau_{j+1}/n_{j+1}$

In this case, when the last piece of stage-j initial WIPs arrives at stage j+1, it has to wait. Let the arrival time be t_0. At stage j+1, the n_{j+1} units of assigned machines must have been busy processing its own initial WIPs and/or the flow-in wafers from stage j up to time t_0. It is easy to figure out the time that all the WIP_j pieces of wafers finish both stages j and j+1 simply being

$$T_{j(j+1)} = (\mathrm{WIP}_j + \mathrm{WIP}_{j+1})\tau_{j+1}/n_{j+1}. \qquad (4.3)$$

The calculations above assume that the workload of a piece of wafer could be equally divided among assigned machines. This approximation is good when $\mathrm{WIP}_j > n_j$, which is mostly the case in real applications.

Now extend the ideas of the two-stage cycle time calculation to develop the recursive calculation of cycle times for going through a general number of stages.

General Cycle Time Calculation (k > j+1)

Suppose that cycle times $T_{j(k-1)}$ and $T_{(j+1)k}$ are available for a pair of stages (j, k). There are also two cases for the calculation of cycle time T_{jk}.

Case G1: $T_{j(k-1)} > T_{(j+1)k} + (\mathrm{WIP}_j - 1)\tau_k/n_k$

By definition, $T_{j(k-1)}$ is the time that the last piece of the WIP_j wafers is completed at stage k-1. In this case, all the initial WIPs of stages j to k but the last piece of the WIP_j finish stage k at time $T_{j(k-1)}$. That is, the last piece of WIP_j does not have to wait when it arrives at stage k. The time that all the WIP_j pieces of wafers need to finish stage k is therefore

$$T_{jk} = T_{j(k-1)} + \tau_k/n_k. \qquad (4.4)$$

Case G2: $T_{j(k-1)} \leq T_{(j+1)k} + (\mathrm{WIP}_j - 1)\tau_k/n_k$

Similar to the reasoning for Case 2, when the last piece of stage-j initial WIPs arrives at stage k, i.e., at time $T_{j(k-1)}$, it has to wait in this case. Since $T_{(j+1)k}$ is the time needed to complete the initial WIPs from stages j+1 to k and is shorter than $T_{j(k-1)}$, the time for all wafers of WIP_j to finish stage k is approximated by

$$T_{jk} = T_{(j+1)k} + \mathrm{WIP}_j\tau_k/n_k. \qquad (4.5)$$

The recursion of SOPEA starts with computing $T_{j(j+1)}$ for j = 1,..., J-1 by applying the two-stage formula (Eqs. (4.2) and (4.3)). Cycle times $\{T_{j(j+d)}, j = 1,\ldots, J-d\}$ are then calculated by using the general formula in Eqs. (4.4) and (4.5) with d increased from 2 to J-1. Such a procedure generates all the cycle times T_{jk}'s for $1 \leq j < k \leq J$. Under a given set of target settings, the amount of wafers that may flow into a stage j during a day can be easily computed by adding up the WIPs of stage j's uptream stages that are within 24 hours of cycle times, i.e.,

$$\mathrm{Flow\_in}_j = \sum_{j' \in A_j} \mathrm{WIP}_{j'}, \text{ where}$$

$$A_j \equiv \{j' \mid j' < j-1 \text{ and } T_{j'(j-1)} \leq 24 \text{ hrs}\}. \qquad (4.6)$$

V. Field Implementation Results

Field implementation of TSS has been in phases for daily application to fabs of high variety, low-volume production. Before the implementation of TSS, daily targets were purely determined by experienced engineers assisted by simple spread sheets, which took an engineer about 30 minutes of decision time. In the first phase, there were no MPSC or CE modules. Only were the daily demanded moves of the output stage set manually; demanded moves of other stages were essentially set to zero. Capacity estimates were obtained by using the capacity status at the beginning of the day plus manual adjustments. Flow-in estimation was based on an empirical rule instead of a cycle time based scheme or SOPEA. After the implementation, a production supervisor needs only review the targets generated by TSS and adjust a small portion of them every morning for about 10 minutes. Targets are then delivered to shop floor after further review by the daily production meeting. Comparing the fab performance before and after the implementation (Figures 6.1 and 6.2), average total fab WIP declined by 8% and daily fab total wafer moves increased by nearly 20%.

SOPEA was integrated into TSS in phase 2. Further improvements were immediately observed as shown in Figures 6.2(a)-(c), where the number of daily fab total wafer moves increased by another 5%, and the number of stages that have more than 10% difference between the scheduled and the actual targets was reduced by about 10%. SOPEA allows daily WIP dependent cycle time prediction, which is very difficult for even an experienced fab supervisor in a fab. It was reported that phases 1 and 2 implementations in conjunction with other cycle time reduction efforts at the fab lead to the reduction of average cycle time per layer from 3.25 to 2.96 days and the reduction of average +2 sigma of per-layer cycle time from 4.63 to 3.68.

Phase 3 of TSS implementation has been designed by simplifying the data requirements and algorithmic complications, enriching functionality and enhancing user interface of the phases 1 and 2. By trading off between accuracy and data maintenance effort, statistical cycle times are adopted in flow-in calculation instead of SOPEA because SOPEA requires accurate stage processing times, which are difficult to maintain in a fab.

After phase 3 of field implementation, there have consistently been less then 20 out of a total of 140 production stage targets that require some adjustments each day. Figure 6.3(a) depicts that the scheduled total moves are mostly within 5% difference of what were actually achieved in the shop floor. Since the daily capacity variation of machine is about 10 - 15% of the forecasted capacity, we consider the scheduled daily target of a stage accurate if it is within 15% of the actually achieved number of moves. The performance of TSS phase 3 on stage target accuracy is given by Figure 6.3(b), which shows that about 50% of scheduled stage targets are accurate. Detailed analyses indicate two potential causes of the by-stage inaccuracy. The first is the discrepancy between the demanded moves calculated from MPS and what are desired in a day by the shop floor managers. The second is the use of empirical cycle times. In addition, coarse modeling of key machiness of individual stages may also contribute to the inaccuracy.

VI. Conclusions

The core methodology of designing a daily target setting system (TSS) for semiconductor fabs has been described in this paper. In the methodology, priority of machine capacity allocation is given to meeting the master production schedule over maximizing machine utilization. To smooth the production flow, capacity of a machine group is allocated to individual stages in proportional to their demanded moves. A deterministic queueing analysis-based algorithm, SOPEA, is designed to estimate cycle times and wafer flows. Our methodology iterates between capacity allocation and cycle time estimation until a fixed-point capacity allocation is achieved. This methodology has been implemented into various versions of a target setting system, which has served as a decision support tool for fab operations. Field implementation results demonstrated a quick convergence of the fixed-point iteration, significant reduction in mean WIPs/cycle times and their variances, increase in moves, and improvement in target achievement percentage.

References

[BuF93] R. Burden, J. Faires, *Numerical Analysis*, 5th ed., PWS-KENT Publishing Co., Boston, 1993.

[CCW95] S.-C. Chang, W.-L. Jan, T.-H. Wang, C.-S. Chang, "Analysis of Proportional Machine Allocation Scheme in a Deterministic Reentrant Line," *Proceedings of 1995 International Conference on Emerging Technology of Factory Automation*, Paris France, Oct.10-12,

[CFY94] D. Conors, G. Feigin, D. Yao, "Scheduling Semiconductors Lines Using a Fluid Network Model," *IEEE Transactions on Robotics and Automation*, Vol. 10, 1994, pp.88-98.

[CLP95] S.-C. Chang, L.-H. Lee, L.-S. Pang, T. Chen, Y.-C. Weng, H.-D. Chiang, D. Dai, "Iterative Capacity Allocation and Production Flow Estimation for Scheduling Semiconductor Fabrication," *17th IEEE International Electronics Manufacturing Technology Symposium*, Austin TX, Oct., 1995, pp. 508 – 512.

[GSS96] C. Glassey, J. Shanthikumar, S. Seshadri, "Linear Control Rules for Production Control of Semiconductor Fabs," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 9, No.4, Nov., 1996,pp.536-549.

[Lea94] R. Leachman, "Production Planning and Scheduling Practices Across the Semiconductor Industry," *Technical Report*, ESRC 94-29/CSM-18,Engineering Systems Research Center, U.C. Berkeley, Berkeley, CA, Sept. 1994.

[LCP96] D.-Y.Liao, S.-C.Chang, K.-W.Pei, C.-M.Chang, "Daily Scheduling for R&D Semiconductor Fabrication,"*IEEE Transactions on Semiconductor Manufacturing*, Nov., 1996, pp. 550-561.

[LTC96] Li, S., T. Tang, and D. W. Collins, "Minimum inventory variability schedule with application in semiconductor fabrication," *IEEE Trans. Semi. Manuf.*, 9 (1996), pp.145-149.

[LRK94] S. Lu, D. Ramaseamy P. Kumar, "Effcient Scheduling Policies to Reduce Mean and Variance of Cycle Time in Semiconductor Manufacturing Plants," *IEEE Transaction on Semiconductor Manufacturing*, Vol. 7, No.3, Aug., 1994, pp.374-388.

[Wei88] L. Wein, "Scheduling Semconductor Wafer Fabrication," IEEE Transactions on Semiconductor Manufacturing, Vol. 1, No.3, Aug., 1988, pp.115-130.

[WWT98] G.-L.Wu, K. Wei, C.-Y. Tsai, S.-C. Chang, N.-J. Wang, R.-L. Tsai, H.-P. Liu, "TSS: a Daily Production Target Setting System for Foundry Fabs," *Proceedings of International Symposium on Semiconductor Manufacturing 1998*, Tokyo, Oct. 7-9, 1998, *pp.75~78.*

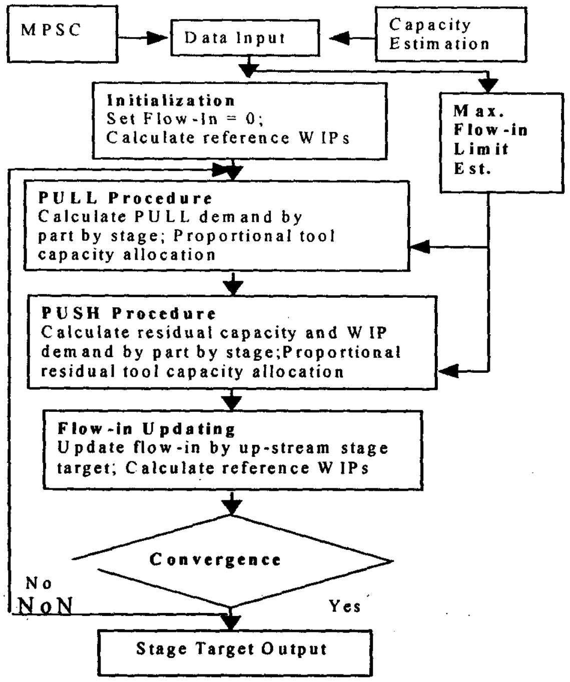
MPSC
Data Input
Capacity Estimation
Initialization
Set Flow-In = 0;
Calculate reference WIPs
Max. Flow-in Limit Est.
PULL Procedure
Calculate PULL demand by part by stage; Proportional tool capacity allocation
PUSH Procedure
Calculate residual capacity and WIP demand by part by stage;Proportional residual tool capacity allocation
Flow-in Updating
Update flow-in by up-stream stage target; Calculate reference WIPs
Convergence
No
NoN
Yes
Stage Target Output

Figure 3.1: TSMF Flow Chart

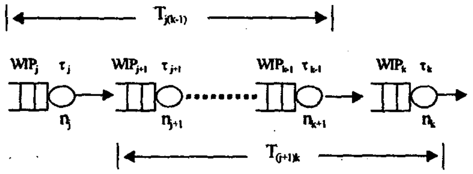

Figure 4.1: Partial Process Flow

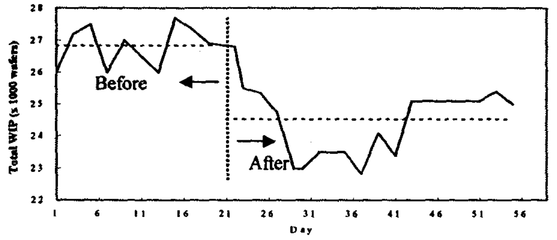

Figure 6.1(a): WIP Profile before and after Phase 1 Implementation

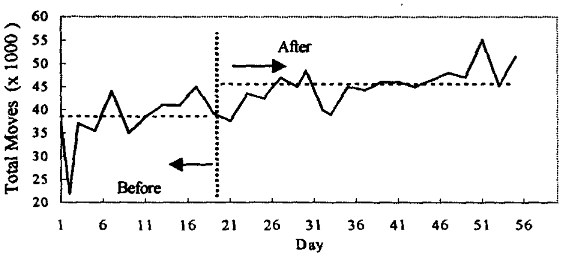

Figure 6.1(b): Profile of Total Moves before and after Phase 1 Implementation

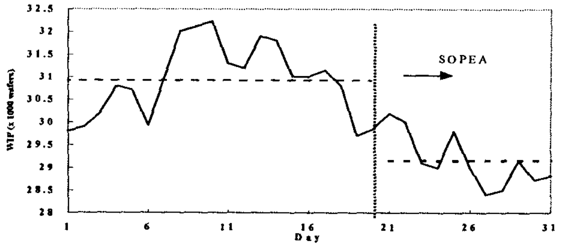

Figure 6.2(a): WIP Profile before and after Phase 2 (SOPEA) Implementation

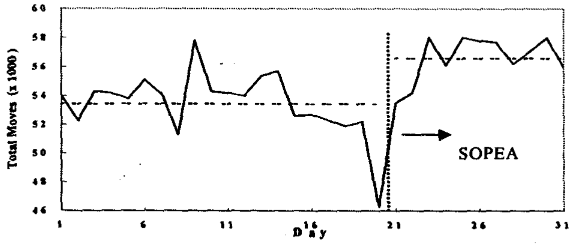

Figure 6.2(b): Profile of Total Moves before and after Phase 2 (SOPEA) Implementation

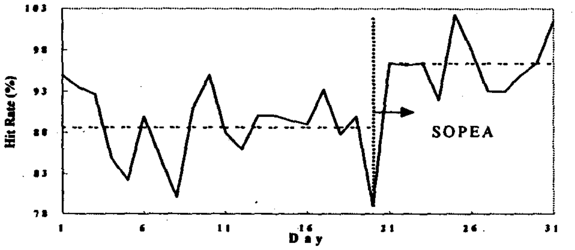

Figure 6.2(c): Target Hit Rate of Total Moves before and after Phase 2 (SOPEA) Implementation

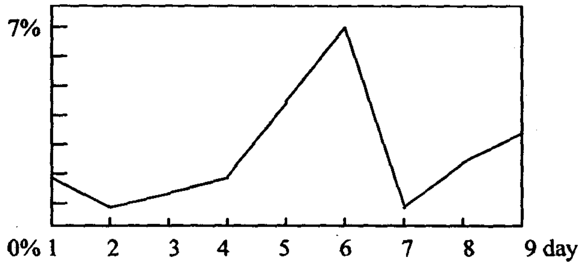

Figure 6.3(a): Difference between Target and Actual Total Moves

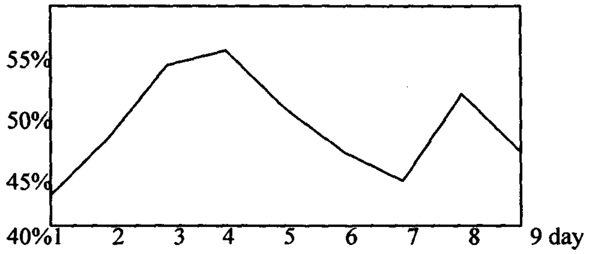

Figure 6.3(b): TSS Performance on Stage Targets